



(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
22.12.1999 Bulletin 1999/51

(51) Int. Cl.⁶: **G06F 17/30**

(21) Application number: **99201873.9**

(22) Date of filing: **14.06.1999**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
 Designated Extension States:
AL LT LV MK RO SI

(72) Inventor: **Lipkin, Efrem**
Berkeley, CA 94703 (US)

(74) Representative:
Hanna, Peter William Derek et al
Tomkins & Co.,
5 Dartmouth Road
Dublin 6 (IE)

(30) Priority: **18.06.1998 US 99489**

(71) Applicant:
SUN MICROSYSTEMS, INC.
Palo Alto, California 94303 (US)

(54) **Client Intermediation of server applications**

(57) One embodiment of the present invention provides a method and an apparatus for providing a client-side intermediary that communicates with an application (112) on a second server computer system (104). The method operates by receiving a composite message at a client computer system (108) from a first server computer system (102), and examining (304) type information from the composite-message. This type information specifies how the composite message is formatted, and can be used to select an application that is capable of processing the composite message. The method uses the type information to look up (306) a network address of the application residing on the second server computer system. This address is used (308) to forward (310) the composite message to the application on the second server computer system. The above embodiment can be implemented within a browser (114) on the client computer system, or within a separate application (120) on the client computer system. A return communication can be facilitated by receiving a return composite message at the client computer system from the second server computer system, and examining type information from the return composite message. This type information is used to look up an address of a source application on the network (106), and this address is used to forward the return composite message to the source application.

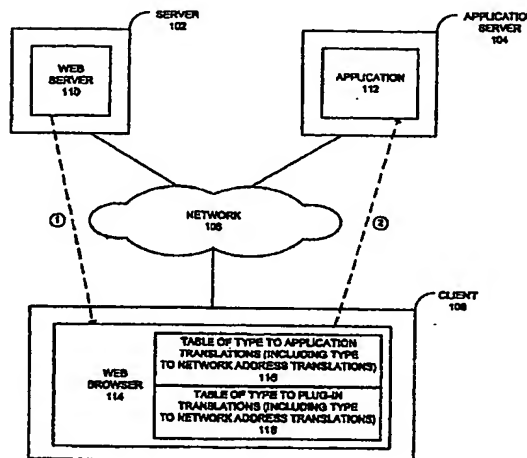


FIG. 2

Description

BACKGROUND

[0001] The present invention relates to client-server architectures in distributed computer systems. More particularly, the present invention relates to a method and an apparatus that operates on a client computing system and implements what appears to be an application located on the client computing system, but which actually acts as a proxy for an application on a remote server computing system.

[0002] The recent proliferation of client-server based distributed system has led to the development of numerous server applications, located on server computer systems, that interact with client applications, located on client computer systems. For example, one recently developed client application is an "electronic wallet," which contains financial instruments in electronic form, such as electronic cash, electronic debit cards or electronic credit cards. An electronic wallet typically resides on a client computer system, and performs financial transactions, such as purchases, by communicating with a server application on a server computer system. For example, a user on a client computer system might decide to purchase software from a server computer system that belongs to a software vendor. Protocols such as the Secure Electronic Transactions (SET) protocol and the Open Trading Protocol (OTP) enable the server computer system to receive a payment for the software from an electronic wallet on the client computer system. Using these protocols, this payment is automatically deducted from an account linked to the electronic wallet on the client computer system, and is automatically credited to an account linked to an application on the server computer system.

[0003] One problem with many existing server-based applications, as well as with protocols such as SET and OTP, is that they are designed to interact with a client application that resides on a client computer system. Locating an application, such as an electronic wallet, on a client computer system has certain disadvantages. First, an owner of the client application may want to use the client application from a number of different client computer systems. In this case, if the client application is tied to a particular client computer system, the owner of the client application cannot access the client application from another client computer system. Second, installing a client application, such as a wallet, on a client computer system can take up storage space on the client computer system and may require additional maintenance on the client computer system -- to update the client application, for example. Additionally, retrieving code from a server computer system can incur a long delay in downloading the code from the server computer system.

SUMMARY

[0004] The present invention allows a client application to interact with existing server applications and protocols, and allows the owner of the client application to use the client application from different client computer systems. One embodiment of the present invention provides a method and an apparatus for providing a client-side intermediary that communicates with an application on a second server computer system. The method operates by receiving a composite message at a client computer system from a first server computer system and examining type information from the composite message. This type information specifies how the composite message is formatted, and can be used to select an application that is capable of processing the composite message. This type information is used to look up an access mechanism that can be used to access the application residing on the second server computer system. This access mechanism is used to forward the composite message to the application on the second server computer system. The above embodiment can be implemented within a browser on the client computer system, or within a separate application on the client computer system. A return communication can be facilitated by receiving a return composite message at the client computer system from the second server computer system, and examining type information from the return composite message. This type information is used to look up an access mechanism for a source application on the first server computer system, which is used to forward the return composite message to the source application, receiving commands from a user at a client computer system.

[0005] Thus, the present invention allows an application on a first server to communicate with what appears to be an application located on a client computer system. In reality, an intermediary on the client computer system forwards the communications to an application on a second server. This allows a client application program, such as an electronic wallet, to be located on an accessible server on a network. This makes it possible for the owner of an application program, such as a wallet, to use the application program from different client computer systems. It additionally frees the client computer system from the burden of storing the application program.

BRIEF DESCRIPTION OF THE FIGURES

[0006]

FIG. 1 illustrates a distributed computer system in accordance with a first embodiment of the present invention.

FIG. 2 illustrates a distributed computer system in accordance with a second embodiment of the present invention.

FIG. 3 is a flow chart illustrating some of the operations involved in forwarding communications to an application on a remote server accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

[0007] The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

[0008] The data structures described in this detailed description are typically stored on a computer readable storage medium, which may be any device or medium that can store code and/or data for use by a computer system. This includes, but is not limited to, magnetic and optical storage devices such as disk drives, magnetic tape, CDs (compact discs) and DVDs (digital video discs), and computer instruction signals embodied in a carrier wave.

Description of One Embodiment

[0009] FIG. 1 illustrates a distributed computer system in accordance with a first embodiment of the present invention. The embodiment illustrated in FIG. 1 includes server 102 coupled to server 104 and client 108 through network 106. Network 106 generally refers to any type of wire or wireless link between computers, including, but not limited to, a local area network, a wide area network, or a combination of networks. In one embodiment of the present invention, network 106 includes the Internet. In another embodiment, network 106 includes multiple networks, including a local area network and the Internet. Client 108 may include any node on a computer network including computational capability and including a mechanism for communicating across network 106. Servers 102 and 104 may include any node on a computer network including computational capability, and possibly data storage capability, as well as a mechanism for servicing requests from a client for computational or data storage resources.

[0010] In the illustrated embodiment, server 102 includes web server 110. Web server 110 is an application program residing on server 102 that facilitates the presentation of at least one website to clients of server 102. To this end, web server 102 presents inter-linked pages of text and graphical images and programs to be accessed by browsers on client systems. (For purposes of this detailed description, a browser is a program, typ-

ically located on a client computer system, which allows a user to navigate through collections of data, such as web sites, on remote computer systems.) In the illustrated embodiment, web server 110 is accessible from any browser coupled to the World Wide Web via the Internet. Although the illustrated embodiment operates in the context of a web server 110 and a web browser 114, the present invention will work for numerous types of client programs and server programs, and is not limited to just web servers and web browsers.

[0011] Application server 104 contains application 112, which receives communications from client 108, and performs operations that appear to be performed by an application located on client 108. In one embodiment of the present invention, application 112 includes code and data to implement an electronic wallet on application server 104. (For purposes of this detailed description, an electronic wallet is a collection of financial instruments, such as credit cards, debit cards and cash in electronic form, possibly stored on a smart card. An electronic wallet is typically implemented through an application program that performs transactions on the financial instruments.) In general, application 112 may implement any type of application, not only an electronic wallet. Note that application 112 can receive communications from other clients besides client 108. This allows the owner of an application, such as a wallet, to use the application from different clients. It also allows the server to act as a client for multiple users.

[0012] Client 108 includes web browser 114, mediator 120 and table 122. Web browser 114 includes code and data that allows a user on client 108 to navigate through web pages located on servers distributed across network 106. One example of such a browser is Netscape Navigator 3.0, produced by the Netscape Communications Corporation of Mountain View, California. In the illustrated embodiment, web browser 114 contains table 116, which includes translations from type information to identifiers for applications, as well as table 118, which includes translations from type information to identifiers for plug-ins. (For purposes of this detailed description, a plug-in is executable code that can be added to an application program that is able to share a window with the application program.) Tables 116 and 118 may take the form of a Helper Application Table, which is a table inside of a browser that maps type information for a composite message to application programs that are able to process the composite message.

[0013] Mediator 120 is an application program, or a plug-in that forwards communications to application 112 on application server 104. Mediator 120 communicates with table 122, which contains translations from type information to addresses for remote applications, such as application 112 on application server 102.

[0014] The system illustrated in FIG. 1 operates as follows. First, a user operating web browser 114 on client 108 initiates some type of transaction with web server 110 on server 102. In response, web server 110 sends

a composite message across network 106 to web browser 114. (For purposes of this detailed description, a composite message can be any type of data transfer across a network that includes one or more transmissions. A composite message may include, but is not limited to, a single document, a collection of documents, a single message, a multi-part message, a single file or a set of files.) Next, web browser 114 reads type information from the composite message in order to determine what type of composite message has been received from web browser 114. This type information may be determined by looking at an extension on the filename for a file associated with the composite message (For example a ".txt" extension may indicate a file containing ASCII text). Alternatively, the type information may be read from a composite message header, such as a Multipurpose Internet Mail Extensions (MIME) type header. (MIME is a standard for type information appended to a composite message, including information specifying how information within the composite message is formatted and/or partitioned.) Next, the type information is used to lookup an application or a plug-in that is able to process the composite message. This lookup may be performed by accessing table 116, which contains type-to-application translations, or by accessing table 118, which contains type-to-plugin translations.

[0015] If the type information specifies a client application, such as a wallet, with an implementation on a remote server, the composite message is sent to mediator 120. Mediator 120 uses the type information to look up an address of the application on the remote server in table 122, which contains type-to-address translations. In one embodiment of the present invention, this address of a remote application includes a Universal Resource Locator (URL). The composite message is then forwarded to the application on the remote server. Optionally, the composite message can be forwarded within an appropriate protocol to it to make it appear as if it originated from a client application on client 108.

[0016] Alternatively, if the type information specifies a client application that exists on client 108, the composite message is sent to the client application directly.

Description of Another Embodiment

[0017] FIG. 2 illustrates a distributed computer system in accordance with a second embodiment of the present invention. In this embodiment, instead of using a separate mediator 120 to forward the composite message to application 112 as in FIG. 1, web browser 114 performs the forwarding itself. In this embodiment, web browser 114 is modified so that it can store type-to-address translations in tables 116 and 118. If the type information specifies a client application, such as a wallet, with an implementation a remote server, the corresponding entry in either table 116 or 118 contains a remote address for the application. In this case, the composite message is sent to the remote address. Alternatively, if

the type information specifies a client application that exists on client 108, the composite message is sent to the client application directly.

Method of Operation

[0018] FIG. 3 is a flow chart illustrating some of the operations involved in forwarding communications to an application on a remote server accordance with an embodiment of the present invention. The system starts at state 300 and proceeds to state 302. In state 302, the system receives a composite message. In the embodiment illustrated in FIG. 1, this corresponds to web browser 114 in FIG. 1 receiving a composite message from web server 110 on server 102. This composite message can be in response to operations performed by a user of client 108 in accessing web server 110 through browser 114. Next, the system proceeds to state 304. In state 304, the system reads type information from the composite message. This type information specifies how the composite message is formatted, and thereby implicitly specifies which application programs can read the composite message. This type information may be determined by reading type information, which accompanies the composite message, such as MIME-type information. Alternatively, the type information may be determined by reading a filename extension for a file associated with the composite message. The system next proceeds to state 306.

[0019] In state 306, this type information is used to look up an address of an application on a remote server. In the embodiment illustrated in FIG. 1, this corresponds to mediator 120 looking up an address in table 122. In the embodiment illustrated in FIG. 2, this corresponds to web browser 114, looking up an address in either table 116 or table 118. The system next proceeds to state 308.

[0020] In state 308, the system puts the composite message in appropriate form for a second application. This may include modifying the composite message to make the composite message appear as though it originated from a client computer system, instead of from web server 110. It may also include modifying the composite message to make the composite message appear as though it originated from another server. In one embodiment of the present invention, this is accomplished by prepending http information onto the composite message. Next, the system proceeds to state 310. In state 310 the system forwards the composite message to the address that was looked up in state 306. In the embodiments illustrated in FIGs. 1 and 2, this corresponds to the composite message being forwarded across network 106 to application 112 on application server 104.

[0021] Although the above-illustrated embodiment of the present invention illustrates a limited case where communications between a first server computer system and a client computer system are redirected to a

second server computer system, these interactions can be generalized to larger numbers of server computer systems. For example, communications to an application on the client computer system can be redirected multiple server computer system, wherein the multiple server computer systems each handle portions of the client application program.

[0022] The foregoing descriptions of embodiments of the invention have been presented for purposes of illustration and description only. They are not intended to be exhaustive or to limit the invention to the forms disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art.

Claims

1. A method for providing a client-side intermediary that communicates with an application (112) on a second server computer system (104), comprising:

receiving commands from a user at a client computer system (108);
 allowing the user to interact with a first server computer system (102) through the client computing system (108);
 in response to interactions by the user, receiving (1, 302) a communication directed to the client computer system (108) from the first server computer system (102);
 examining (304) the communication to determine what type of information is contained in the communication; and
 if the type of information in the communication can be processed by an application (112) on the second server computer system (104), forwarding (2, 4, 310) the communication to the application (112) on the second server computer system (104).

2. The method of claim 1, comprising:

receiving (1, 302), at a client computer system (108), a communication as a composite message from a first server computer system (102) directed to the client computer system;
 examining (304) type information from the composite message, the type information specifying how the composite message is formatted;
 using (306, 308) the type information to engage an access mechanism (114, 120) through which the application (112) on the second server computer system (104) can be accessed; and
 forwarding (2, 4, 310) the composite message to the application (112) on the second server computer system (104) using the access mechanism.

3. The method of claim 2, further comprising:

receiving, at the client computer system (108), a return composite message from the second server computer system (104);
 examining return type information from the return composite message;
 using the return type information to engage a source access mechanism through which a source application (110) on the first server computer system (102) can be accessed; and
 forwarding the return composite message to the source application (110) on the first server computer system using the source access mechanism.

4. The method of claim 2, wherein:

receiving the composite message includes receiving the composite message at a browser (114) on the client computer system (108); and
 using the type information to engage the access mechanism includes using Multipurpose Internet Mail Extensions (MIME) type information to look up an entry in a helper application table (116, 122) in the browser (114), the entry in the helper application table including the address of the application on the second server computer system (104).

5. The method of claim 2, wherein:

receiving the composite message includes receiving (1) the composite message at a client-side application (120) on the client computer system; and wherein forwarding the composite message to the application (112) on the second server computer system (104) includes forwarding (2) the composite message from the client-side application (120) to the application on the second server computer system (104).

6. The method of any one of claims 1 to 5, wherein the type information includes MIME-type information.

7. The method of any one of claims 1 to 5, wherein the type information is specified by a filename extension for a file associated with the communication or composite message.

8. The method of any one of claims 1 to 7, wherein forwarding the communication or composite message to the application (112) on the second server computer system (104), includes repackaging the composite message.

9. The method of claim 8, wherein repackaging the

composite message makes the composite message appear to originate from the client computing system.

10. The method of claim 8, wherein repackaging the composite message makes the composite message appear to originate from a third server computing system.
11. The method of any one of claims 2 to 5, wherein engaging the access mechanism (114, 120) includes referencing the application (112) on the second server computer system (104) through a Universal Resource Locator (URL).
12. The method of any one of claims 1 to 11, wherein the application (112) implements an electronic wallet.
13. A computer readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for providing a client-side intermediary that communicates with an application (112) on a second server computer system (104), comprising:

receiving (1, 302), at a client computer system (108), a composite message from a first server computer system (102) directed to the client computer system (108);
examining (304) type information from the composite message, the type information specifying how the composite message is formatted;
using (306, 308) the type information to engage an access mechanism (114, 120) through which the application (112) on the second server computer system (104) can be accessed; and
forwarding (2, 4, 310) the composite message to the application (112) on the second server computer system (104) using the access mechanism.

14. An apparatus that provides a client-side intermediary that communicates with an application (112) on a second server computer system (104), comprising:

a receiving means (302) that receives a composite message from a first server computer system (102) directed to a client computer system (108);
a reading means (304), in communication with the receiving means (302), that reads type information from the composite message, the type information specifying how the composite message is formatted;

an access means (306, 308), in communication with the reading means (304), that uses the type information to trigger an access to the application on the second server computer system; and

a forwarding means (310), in communication with the receiving means (302), that forwards the composite message to the application (112) on the second server computer system (104) using the access means.

15. The apparatus of claim 14, wherein:

the receiving means is configured to receive a return composite message from the second server computer system (104);
the reading means is configured to read return type information from the return composite message;
the access means is configured to use the return type information to trigger an access to a source application (110) on the first server computer system (102); and
the forwarding means is configured to forward the composite message to the source application on the first server computer system using the access means.

16. The apparatus of claim 14, further comprising a browser (114) on the client computer system (108), the browser including the receiving means (302), the reading means (304), the access means (306, 308) and the forwarding means (310).
17. The apparatus of claim 14, further comprising a client-side application (120) on the client computer system, the client-side application including the receiving means (302), the reading means (304), the access means (306, 308) and the forwarding means (310).
18. The apparatus of any one of claims 14 to 17, wherein the type information includes Multipurpose Internet Mail Extensions (MIME) type information.
19. The apparatus of any one of claims 14 to 17, wherein the type information is specified by a filename extension of a file associated with the composite message.
20. The apparatus of any one of claims 14 to 19, wherein the forwarding means (310) is configured to repackaging the composite message.
21. The apparatus of any one of claims 14 to 20, wherein the access means (306, 308) can reference the application (112) on the second server computer system (104) through a Universal

Resource Locator (URL).

22. The apparatus of any one of claims 14 to 21, wherein the application (112) implements an electronic wallet.

23. A computer system that provides a client-side intermediary that communicates with an application (112) on a second server computer system (104), comprising:

a client computer system (108);
 a first server computer system (102);
 a network (106) coupled to the first server computer system (102), the second server computer system (104) and the client computer system (108);
 a receiving mechanism (302), within the client computer system (108), that receives a composite message from the first server computer system (102) directed to the client computer system (108);
 a reading mechanism (304), in communication with the receiving mechanism (302), that reads type information from the composite message, the type information specifying how the composite message is formatted;
 an access mechanism (306, 308), in communication with the reading mechanism (304), that uses the type information to trigger an access to the application (112) on the second server computer system (104); and
 a forwarding mechanism (310), in communication with the receiving mechanism (302), that forwards the composite message to the application (112) on the second server computer system (104) using the access mechanism.

24. A data structure contained in a computer readable storage medium that facilitates forwarding communications from an intermediary on a client computer system (108) to an application (112) on a remote server computer system (104), comprising:

an indexing structure that takes as input type information specifying how a communication as a composite message is formatted; and
 a plurality of entries (116, 118, 122) accessed by the indexing structure, wherein at least one entry from the plurality of entries facilitates an access to the application (112) on the remote server computer system (104).

25. The data structure of claim 24, wherein the access mechanism makes use of a Universal Resource Locator (URL).

26. The data structure of claim 24, wherein the type

information includes Multipurpose Internet Mail Extensions (MIME) type information.

27. The data structure of claim 24, wherein the type information is specified by a filename extension for a file associated with the composite message.

28. A computer program encoding a set of computer instructions for providing a client-side intermediary that communicates with an application on a second server computer system, which when running on a computer is adapted to perform the method of any one of claims 1 to 12.

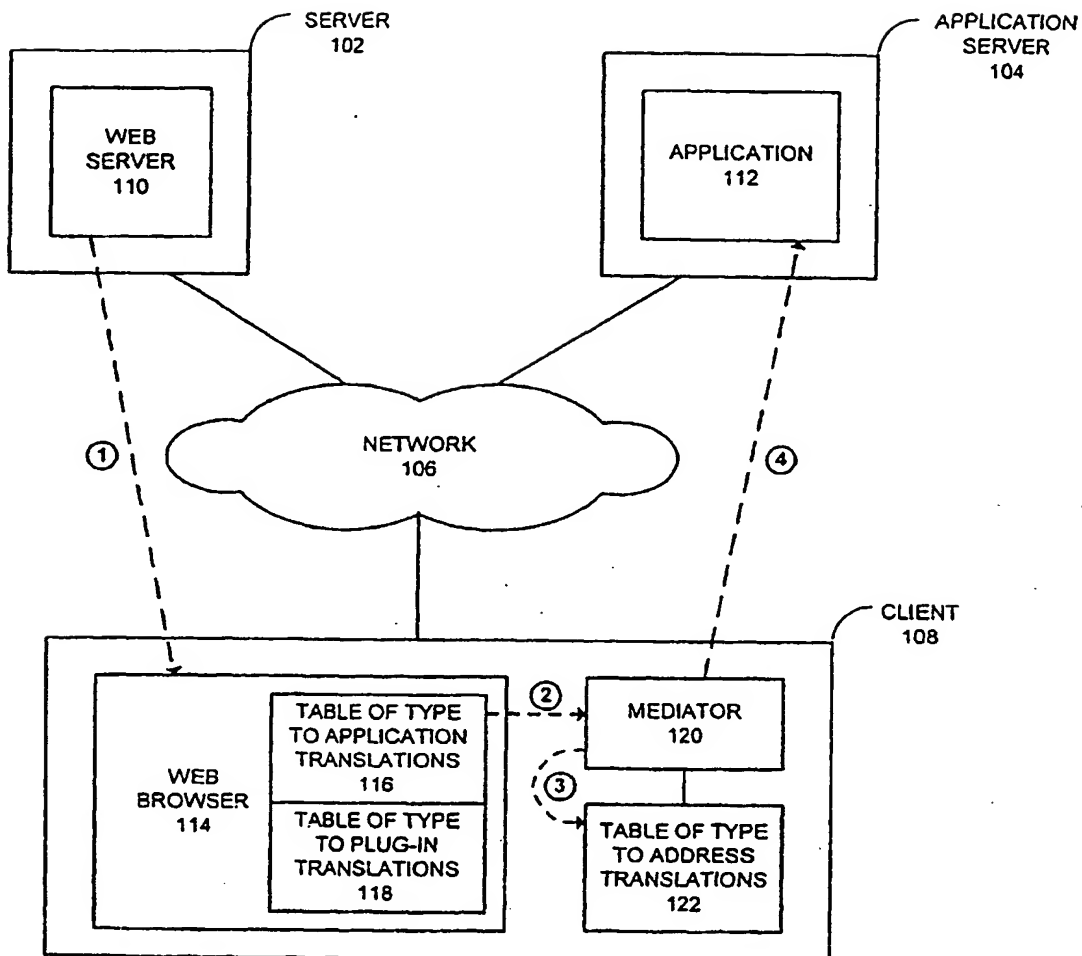


FIG. 1

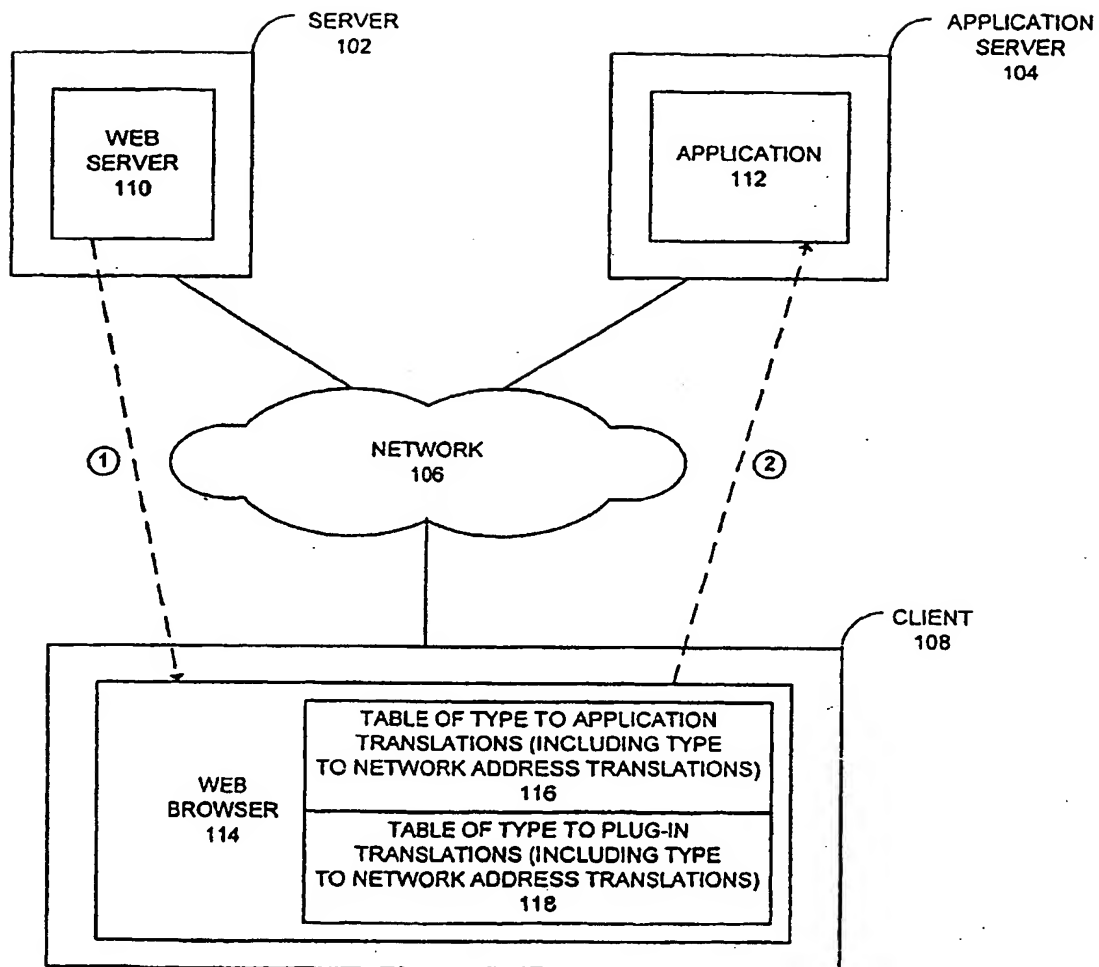


FIG. 2

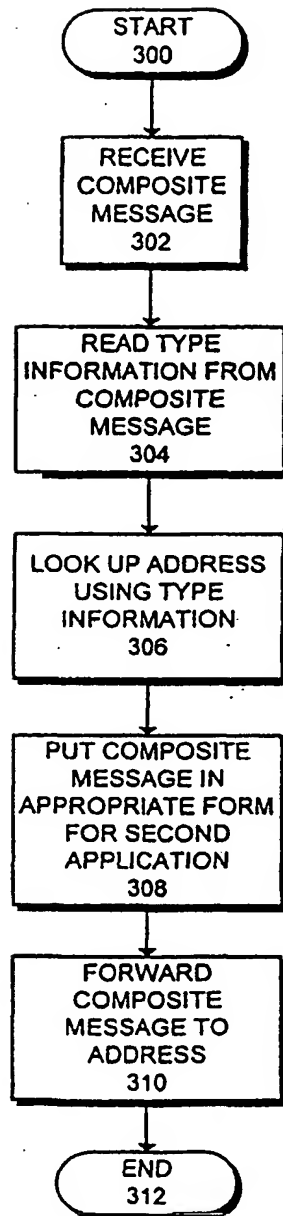


FIG 3.